Analyse de concepts formels pour la construction d'ontologies à partir de textes : la question du corpus

Thibault Mondary, Sylvie Després

Analyse de concepts formels pour la construction d'ontologies à partir de textes : la question du corpus

Thibault Mondary*, Sylvie Després*

*LIPN - UMR 7030
CNRS - Université Paris 13
F-93430 Villetaneuse, France
prenom.nom@lipn.univ-paris13.fr,

Résumé. Nous nous intéressons à l'évaluation d'une méthode permettant l'acquisition de hiérarchies à partir de textes. L'analyse de concepts formels semble donner de bons résultats dans les travaux de Cimiano et al. (2005) et plus récemment Bendaoud et al. (2007). Nous montrons que sur certains corpus les résultats semblent moins utilisables. Est-ce dû au style du texte ? Au domaine à modéliser ? Après une présentation de la méthode utilisée pour construire nos contextes formels, nous analysons les résultats obtenus sur différents corpus. Plusieurs variantes de construction des contextes formels sont proposées, axées sur la sélection des objets, avec pour objectif d'améliorer l'utilisabilité du treillis dans un contexte de conceptualisation semi-automatique.

1 Introduction

Le recours aux textes pour la construction d'ontologies est légitimé par l'hypothèse qu'ils représentent des connaissances partagées stabilisées d'un domaine. Ils ne se substituent pas à un expert du domaine, mais peuvent lui faciliter la tâche de modélisation s'ils sont accompagnés d'outils qui rendent saillantes les informations désirées. Dans cet article nous nous intéressons à l'évaluation des résultats obtenus en utilisant l'analyse de concepts formels (ACF), dans la lignée de (Cimiano et al., 2005) et (Bendaoud et al., 2007). L'ACF est ici destinée à aider le travail de conceptualisation par l'expert lors de la construction d'une ontologie. Les concepts formels et leurs relations de subsomption sont considérées comme des suggestions de concepts et de relations de hiérarchie pour l'ontologie. Le but est d'obtenir l'ACF la plus "suggestive" possible, c'est-à-dire celle qui constitue le meilleur guide pour le travail de conceptualisation. Après avoir rappelé les principes de l'ACF appliquée au texte, nous présentons les résultats des expérimentations menées sur des corpus de genres et de styles différents. Dans une troisième partie nous tentons de déterminer des conditions d'applicabilité de la méthode, et proposons des pistes pour remédier aux problèmes rencontrés. Ceux-ci sont notamment la taille des treillis, le manque de relations de subsomption utilisables, la présence de bruit dans les objets et les attributs du contexte formel qui nuisent à l'utilisabilité du treillis comme guide pour le travail de conceptualisation.

2 Analyse de concepts formels appliquée aux textes

L'analyse de concepts formels (Ganter et Wille, 1999) est une méthode de classification symbolique qui vise à découvrir tous les regroupements possibles d'éléments ayant des traits en commun. La hiérarchie résultante qui regroupe des objets partageant les mêmes propriétés est appelée treillis de concepts. La notion centrale de l'ACF est le **contexte formel**. C'est un triplet $\mathbb{K} = (G, M, I)$ où G est un ensemble d'objets, M un ensemble d'attributs et I une relation binaire entre G et M appelée relation d'incidence de $\mathbb{K}$ et vérifiant $I \subseteq G \times M$. Un couple $(g, m) \in I$ signifie que l'objet $g \in G$ possède l'attribut $m \in M$. Soit $\mathbb{K}$ un contexte formel. Pour tout $A \subseteq G$ et $B \subseteq M$, on définit $A' = \{m \in M | \forall g \in A, (g, m) \in I\}$ et $B' = \{g \in G | \forall m \in B, (g, m) \in I\}$. A' est l'ensemble des attributs communs à tous les objets de A et B' est l'ensemble des objets possédant tous les attributs de B. Nous pouvons maintenant définir un **concept formel** comme étant un couple (A, B) tel que $A \subseteq G$ et $B \subseteq M$, vérifiant $A' = B$ et $B' = A$. A est l'**extension** du concept formel et B son **intension**. L'ensemble des concepts formels associés au contexte $\mathbb{K} = (G, M, I)$ est noté $\mathfrak{B}(G, M, I)$. Les concepts de $\mathfrak{B}(G, M, I)$ sont ordonnés par une relation de subsomption $\sqsubseteq$ telle que $(A1, B1) \sqsubseteq (A2, B2) ssi A1 \subseteq A2$, ou de façon duale $B2 \subseteq B1$.

Cimiano et al. (2005) et ultérieurement Bendaoud et al. (2007) proposent de construire le contexte formel en s'appuyant sur les idées de Hindle (1990). En partant d'une analyse syntaxique des phrases obtenue automatiquement à l'aide

	paralyser	paralyser-ABLE	refuser	refuser-ABLE
grève	×			×
entreprise		×		
travailleur			×	
dirigeant			×	
travail				×

TAB. 1 – *Contexte formel exemple*

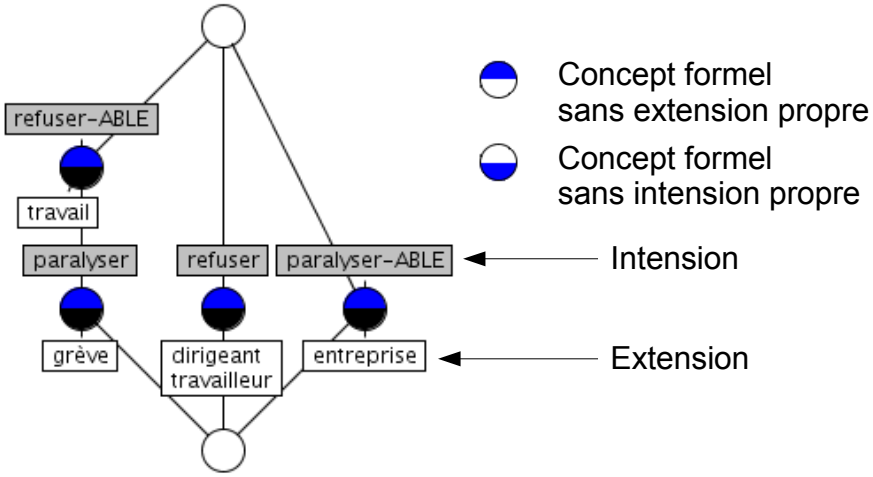

FIG. 1 – *Treillis exemple*

d'un analyseur syntaxique, ils construisent le contexte formel en extrayant de cette analyse les noms et les verbes. Les objets du contexte formel sont les noms apparaissant en tant que sujet, complément d'objet ou locution prépositionnelle. Les attributs du contexte formel sont les verbes de la phrase considérée, complétés du suffixe -ABLE lorsqu'ils proviennent d'un complément ou d'une locution.

Prenons par exemple le texte suivant : *Une grève paralyse l'entreprise. Les travailleurs refusent le travail dominical. Les dirigeants refusent la grève.* En ne gardant que la tête lemmatisée des sujets et des compléments, nous obtenons le contexte de la table 1. Le treillis résultant est présenté figure 1, il est destiné à l'expert. Ce dernier étiquette les concepts formels qu'il juge pertinents pour le domaine.

3 Expérimentations

Pour analyser l'applicabilité de l'ACF, nous considérons trois corpus. Les deux premiers corpus sont composés de documents en français rédigés par l'Organisation Internationale du Travail[1] : les conventions C87, C98 et leurs jurisprudences. Les conventions internationales du travail ont le statut juridique de traités internationaux. C87 s'intitule "Convention sur la liberté syndicale et la protection du droit syndical, 1948" et C98 est la "Convention sur le droit d'organisation et de négociation collective, 1949". Nous distinguons le corpus C87_C98 composé uniquement des deux conventions, 3605 mots, et le corpus C87_C98_Jurisp incluant également les jurisprudences, 99966 mots. Le troisième corpus est constitué de recettes de cuisines récoltées sur internet, en français également, pour 3644 mots.

La figure 2 montre le processus de construction du contexte formel. Du corpus nous extrayons en parallèle les candidats termes et les relations de dépendance syntaxique. L'analyseur syntaxique utilisé pour le français dans ce travail est Syntex (Bourigault et al., 2005). Il a obtenu de bons résultats lors de la campagne EASY[2] et fournit une analyse syntaxique en dépendances bien adaptée à notre objectif de repérage des sujets/compléments, sans utiliser de connaissances externes. YaTeA (Sophie Aubin (2006)) est l'extracteur de termes choisi. Il exploite à la fois des patrons d'extraction et une désambiguïsation endogène par la redondance pour extraire des groupes nominaux candidats-termes.

[1] http://www.ilo.org/ilolex/french/
[2] http://w3.erss.univ-tlse2.fr:8080/index.jsp?perso=bourigault&subURL=syntex.html

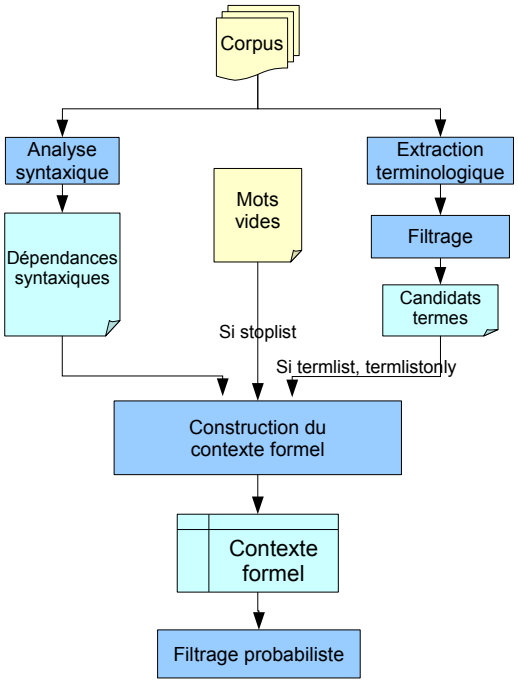

FIG. 2 – *Construction du contexte formel*

Nous construisons le contexte avec les dépendances sujet-verbe(+préposition si elle existe) et complément-verbe(+préposition si elle existe). Nous proposons six variantes de construction du contexte formel, qui diffèrent principalement sur le choix des objets à inclure dans le contexte. L'intuition sous-jacente est que les termes sont de meilleurs descripteurs sémantiques que les mots, mais qu'il y a plus de redondance (et donc plus de regroupements possibles) avec les mots.

usuelle : Inclure dans le contexte formel en tant qu'objets les têtes lemmatisées des sujets et des compléments d'objet et en tant qu'attributs leurs verbes associés (munis des prépositions le cas échéant), à condition que ces têtes soient des noms.

stoplist : Idem à la méthode usuelle, à condition que les candidats objets et attributs n'appartiennent pas à une liste de mots vides fournie par l'utilisateur et ne contiennent pas de chiffre.

termlist : Idem à la méthode usuelle, en tentant d'apparier les candidats objets avec une liste de termes lemmatisés fournie en entrée par l'utilisateur. Si l'appariement échoue, inclure seulement la tête. Par exemple dans l'exemple de la première partie, l'objet "travail" serait remplacé par "travail dominical".

termlist+stoplist : Comme ci-dessus, mais en rejetant au préalable (avant de tenter un appariement) les candidats objets et attributs qui appartiennent à une liste de mots vides ou qui contiennent des chiffres.

termlistonly : N'inclure dans le contexte que les sujets ou compléments d'objet qui s'apparient avec un terme de la liste fournie en entrée. Les verbes associés aux sujets ou compléments sont inclus dans le contexte comme attributs.

termlistonly+stoplist : Comme ci-dessus, mais filtrer les verbes selon une liste de mots vides. Les candidats attributs qui appartiennent à cette liste ou qui contiennent des chiffres sont rejetés.

Dans les méthodes utilisant la liste de termes en entrée une étape de filtrage de la liste de candidats termes est nécessaire. Cette étape est réalisée à l'aide de patrons (dans les expériences, nous utilisons notamment les patrons "supprimer les candidats termes qui contiennent un nombre", ceux qui contiennent "paragraphe ou article"...). La fabrication des patrons nécessite une étape de lecture manuelle des candidats-termes qui est moins coûteuse en temps qu'un filtrage exhaustif. Pour les recettes, nous éliminons 37 candidats-termes, ceux contenant des nombres (par exemple "ébullition 1 l"). Pour les corpus C87_C98, 13 candidats-termes sont exclus. Les patrons appliqués au corpus C87_C98_Cases nous suppriment 767 candidats-termes.

Un filtrage statistique probabiliste *a posteriori* est prévu dans la méthode, mais n'est pas appliqué dans les expériences présentées ici. Dans l'état actuel de notre implémentation nous reprenons celui utilisé par Cimiano et al. (2005), à savoir la probabilité conditionnelle $P(obj|attr) = f(obj, attr)/f(attr)$ pour chaque couple $(obj, attr)$ du contexte formel. $f(obj, attr)$ représente le nombre d'occurrences du couple $(obj, attr)$ dans le contexte[3] et $f(attr)$ le nombre d'occur-

[3]Ce n'est pas forcément une valeur booléenne, mais pour le calcul des concepts formels cela en devient une

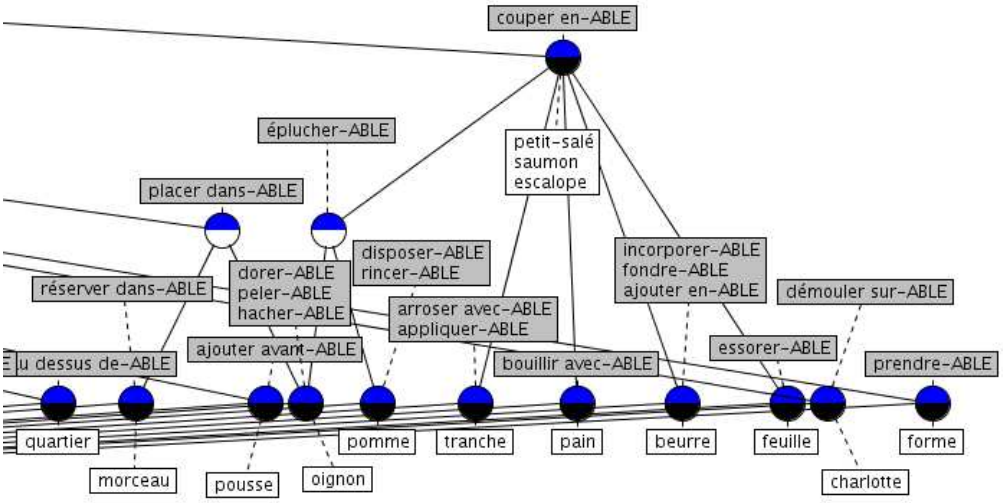

FIG. 3 – *Treillis pour les recettes, méthode stoplist*

rences de l'attribut $attr$. Ce calcul donne pour chaque couple une probabilité, dont il reste à déterminer un seuil pour décider si ce couple doit être conservé dans le contexte formel filtré.

La table 2 donne une analyse quantitative de l'application de l'ACF sur les trois corpus en utilisant la méthode usuelle. Nous pouvons remarquer que le corpus recettes donne deux fois plus de concepts que le corpus C87_C98, pour un nombre de mots similaire. Nous remarquons que le gros corpus C87_C98_Cases n'est pas beaucoup plus productif en concepts que C87_C98, dix fois plus petit. Un aperçu du treillis obtenu pour le corpus recettes est proposé figure 3. La figure 4 montre une partie du treillis pour le corpus C87_C98.

Les figures 5 et 6 présentent une évaluation quantitative de l'impact des variantes de construction sur le treillis. La première figure représente, pour chaque variante, le nombre d'objets, d'attributs et de concepts formels obtenu. Nous pouvons remarquer que les variantes `xxx+stoplist` diminuent dans chaque cas le nombre d'éléments ; pour C87_C98, la diminution engendrée par les mots vides est plus importante que pour les recettes. Ce phénomène est notamment dû à une erreur de l'analyseur syntaxique qui ajoute les numéros des articles des conventions comme objets. Les variantes `termlist(+stoplist)` augmentent le nombre d'objets et de concepts. Ceci est dû au fait que les variantes `usuelles (+stoplist)` travaillent sur des mots et regroupent des termes composés qui ne le seraient pas avec les variantes `termlist(+stoplist)`. Par exemple si dans une phrase nous avons le terme "organisation de travailleurs" en sujet et dans une autre phrase le mot "organisation" en sujet également, la variante `termlist` produira deux objets distincts tandis que les variantes `usuelle(+stoplist)` regroupera les attributs de "organisation de travailleurs" et de "organisation" sous le même objet "organisation". Le nombre d'attributs ne varie pas en passant des variantes `usuelles(+stoplist)` aux variantes `termlist(+stoplist)`, ce qui est normal. Les variantes `termlistonly(+stoplist)`, en incluant uniquement les termes proposés par l'utilisateur, font diminuer notablement tant le nombre d'objets et d'attributs que le nombre de concepts. L'ajout des termes pour le corpus recettes est plus productif en objets que pour le corpus C87_C98, cela s'explique par le fait que les termes sont plus nombreux pour les recettes et s'apparient mieux, c'est-à-dire qu'ils sont davantage présents en tant que sujets ou compléments.

La figure 6 propose une mesure du nombre de concepts selon leur niveau dans chaque treillis. Les concepts de niveau 1 représentent ceux qui sont directement au-dessus de `bottom`. Sur cette figure on observe l'influence de l'injection des termes sur les regroupements : les variantes `termlistonly(+stoplist)` réduisent légèrement la hauteur du treillis. Il est intéressant de noter que pour la méthode usuelle le corpus recettes produit quasiment trois fois plus de concepts de niveau deux que pour le corpus C87_C98. De plus le rapport $Niveau1/Niveau2$ donne 4 pour C87_C98 contre 2.48 pour les recettes, ce qui indique que le treillis du corpus recettes propose quasiment deux fois plus de relations hiérarchiques par objet que l'autre. L'influence des mots vides sur le corpus C87_C98 laisse voir que ses regroupements sont essentiellement vides de sens, ce qui est vrai dans une moindre proportion pour le corpus recettes.

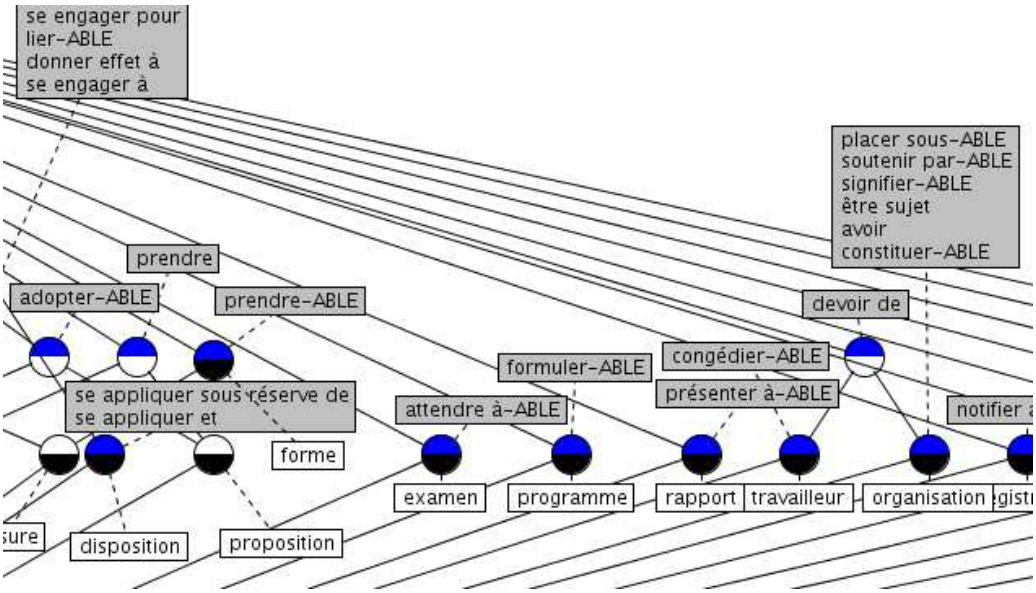

FIG. 4 – *Treillis pour le corpus C87_C98, méthode stoplist*

	C87_C98	C87_C98_Jurisp	Recettes
Nombre d'objets du contexte formel	70	688	122
Nombre d'attributs du contexte formel	74	1379	131
Nombre de concepts formels	54	1564	107
Par rapport au nombre de mots du corpus, base 100	1.5	1.56	2.94
Profondeur du treillis	3	8	3

TAB. 2 – *Comparaison quantitative, méthode usuelle*

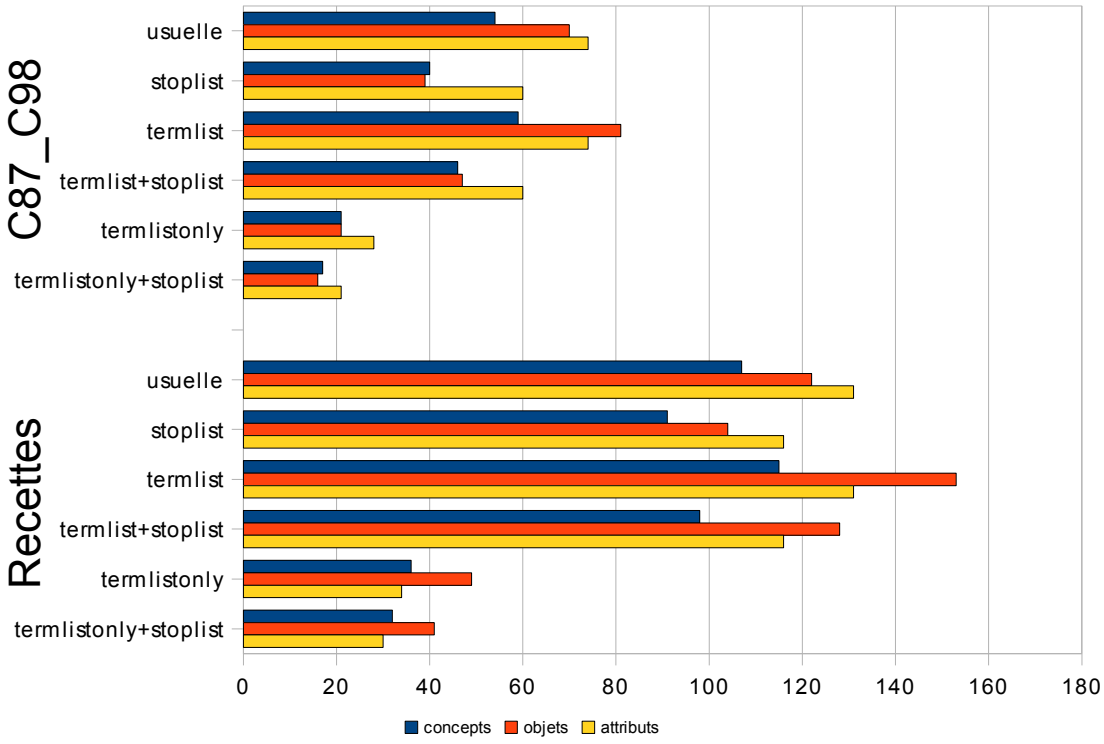

FIG. 5 – *Concepts, objets et attributs, en occurrences pour C87_C98 et Recettes*

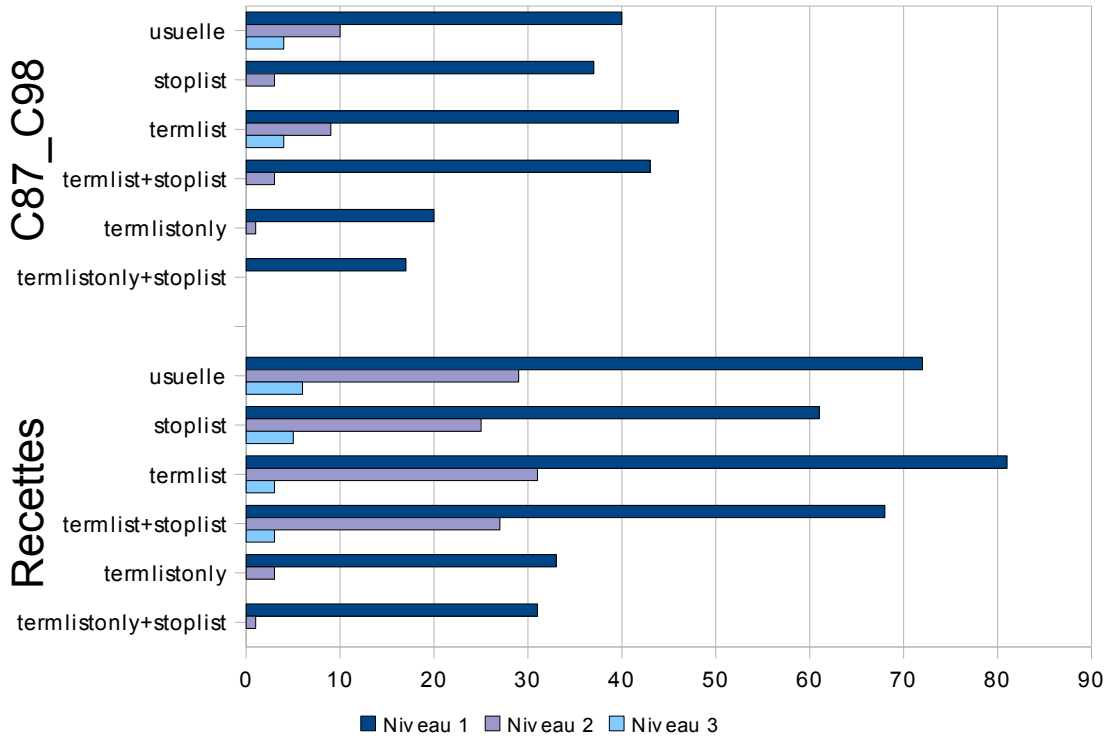

FIG. 6 – *Concepts de niveau 1, 2 et 3 (le maximum), en occurrences pour C87_C98 et Recettes*

	C87_C98	C87_C98_Jurisp	Recettes
Domaine	liberté syndicale	liberté syndicale	cuisine
Nombre de mots	3605	99966	3644
Nombre de phrases	158	4533	299
Phrase la plus longue (en mots)	237	292	58
Nombre moyen de mots par phrase	22,82	22,05	12,18
Nombre de verbes	430	14137	666
Dont modaux	36	1195	6

TAB. 3 – *Caractéristiques des corpus au niveau des occurrences*

4 Interprétations, propositions et conditions d'applicabilité

Le treillis des recettes semble utilisable par l'expert du domaine. On peut y lire que le concept dont l'extension est `{pomme}` a pour intension `{disposer-ABLE, rincer-ABLE, éplucher-ABLE, couper-en-ABLE}`. Il est subsumé par le concept d'intension `{éplucher-ABLE, couper-en-ABLE}` sans extension directe (il n'existe pas dans le texte d'objet ayant uniquement les attributs `éplucher-ABLE` et `couper-en-ABLE`). C'est à l'expert de décider comment étiqueter les concepts formels et quelles relations de subsomption conserver dans l'ontologie, par exemple de déterminer comment interpréter le fait qu'un aliment qui est pelable et hachable soit aussi épluchable et coupable ?

A contrario, le treillis des conventions est très peu utilisable par l'expert : les intensions sont peu informatives (`prendre, placer-sous-ABLE, soutenir-par-ABLE...`) et la seule hiérarchie est constituée de verbes de modalité (en particulier `devoir`) vides de sens. Le treillis du gros corpus comporte sans doute plus de relations intéressantes mais est tellement fourni qu'il en devient illisible et inexploitable.

En considérant les tables 3 et 4, nous remarquons que les phrases du corpus recettes sont en moyenne presque deux fois plus courtes que celles du corpus C87_C98. Empiriquement, l'absence de verbes de modalité (comme "devoir", "pouvoir"...) et la brièveté des phrases semble être un indicateur de la productivité en concepts du treillis. Les phrases du corpus C87_C98 sont beaucoup plus complexes, avec des ellipses et des modalités (par exemple : *Des mesures appropriées aux conditions nationales doivent, si nécessaire, être prises pour encourager et promouvoir le développement et*

	C87_C98	C87_C98_Jurisp	Recettes
Taille du vocabulaire lemmatisé	868	5858	1198
Fréquence moyenne du voc. lemmatisé	4,15	17,06	3,04
Taille du vocabulaire fléchi	1102	9686	1422
Fréquence moyenne du voc. fléchi	3,27	10,32	2,56
Nombre de candidats termes	122	4064	199
Nombre de termes retenus	109	3297	162

TAB. 4 – *Caractéristiques des corpus au niveau du vocabulaire*

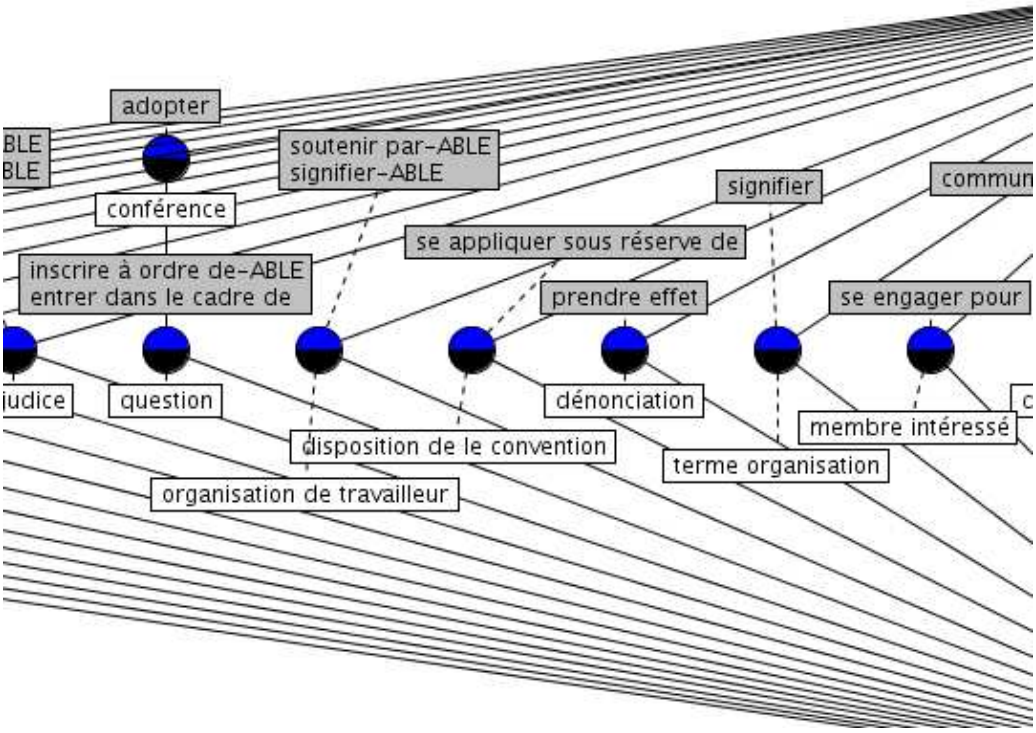

FIG. 7 – *Treillis pour le corpus C87_C98, méthode termlist+stoplist*

l'utilisation les plus larges de procédures de négociation volontaire de conventions collectives entre les employeurs et les organisations d'employeurs d'une part, et les organisations de travailleurs d'autre part, en vue de régler par ce moyen les conditions d'emploi.), ce qui soulève la question de la performance de l'analyseur syntaxique et celle concernant le traitement des modalités : une analyse plus fine que les simples dépendances sujet/verbe/complément semble nécessaire dans ce cas.

Nous avons remarqué après une lecture détaillée du corpus C87_C98 que seulement un tiers du texte est en rapport avec notre tâche de modélisation des violations, le reste servant de cadre à la ratification des conventions. Une ACF sur cette partie donne moins de bruit dans le treillis : il est donc nécessaire de bien sélectionner le corpus que nous voulons utiliser pour modéliser. Cette sélection pourrait être effectuée en analysant le vocabulaire utilisé et en le comparant avec une ressource existante, mais dans le domaine juridique cette procédure est délicate, du fait que les termes du domaine relèvent également du sens commun (par exemple "travail", "organisation") et devrait plutôt être envisagée manuellement, voire à l'aide d'un algorithme d'apprentissage supervisé.

Augmenter la taille du corpus afin d'augmenter la redondance pour extraire plus de concepts est une solution qui nécessite de mettre en place des stratégies de filtrage et d'exploration du treillis. Le filtrage peut s'envisager *a priori* dans la sélection des attributs et des objets intéressants (méthodes `termlistonly(+stoplist)`) ou *a posteriori* en utilisant des méthodes statistiques (comme dans Cimiano et al. (2005)). L'exploration du treillis nécessite des outils permettant une approche incrémentale de la construction, dans laquelle le candidat-treillis est visualisé au moment de la construction du contexte.

Les méthodes de construction `termlistonly` et `termlistonly+stoplist` diminuent le bruit du treillis au détriment des regroupements. Les méthodes `termlist` et `termlist+stoplist` sont celles qui ajoutent de la connais-

sance dans l'interprétabilité des résultats et maximisent les concepts dans le treillis, au détriment toutefois des regroupements (mais moins que pour `termlistonly(+stoplist)`. La figure 7 montre l'augmentation de l'interprétabilité du treillis : l'extension "organisation de travailleur[4]" est plus parlante pour l'expert du domaine que simplement "organisation", idem pour "disposition de le convention".

5 Conclusion et perspectives

Nous avons tenté au travers de cet article de soulever la question de l'applicabilité de l'analyse de concepts formels dans la tâche de construction semi-automatique de hiérarchies ontologiques à partir de textes. Les expériences menées sur des corpus de tailles comparables mais de genres et de styles différents nous ont permis de pointer les limites de l'approche sujet/verbe/complément. La taille des phrases et le nombre de verbes de modalité semblent être des indicateurs de l'interprétabilité du treillis par l'expert. Nous avons proposé une méthode permettant de contenir la taille du treillis et d'augmenter son interprétabilité, mais se pose le problème du coût en temps du filtrage *a priori*. Notre priorité est maintenant d'explorer des méthodes permettant d'optimiser cette étape, en regroupant les synonymes par exemple. Nous nous posons également la question de la pertinence d'inclure dans un même contexte formel les relations sujet/verbe et complément/verbe. Ne sont-elles pas complémentaires pour la construction des concepts qui partagent des attributs de natures différentes ?

Références

Bendaoud, R., M. Rouane Hacene, Y. Toussaint, B. Delecroix, et A. Napoli (2007). Construction d'une ontologie à partir d'un corpus de textes avec l'acf. In F. Trichet (Ed.), *Actes des 18eme journées francophones d'ingénierie des connaissances (IC2007)*. Cépaduès.

Bourigault, D., C. Fabre, C. Frérot, M.-P. Jacques, et S. Ozdowska (2005). Syntex, analyseur syntaxique de corpus. In *Actes des 12èmes journées sur le Traitement Automatique des Langues Naturelles*, Dourdan, France.

Cimiano, P., A. Hotho, et S. Staab (2005). Learning concept hierarchies from text corpora using formal concept anaylsis. *Journal of Artificial Intelligence Research (JAIR) 24*, 305–339.

Ganter, B. et R. Wille (1999). *Formal Concept Analysis*. Berlin : Springer.

Hindle, D. (1990). Noun classification from predicate-argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, Morristown, NJ, USA, pp. 268–275. Association for Computational Linguistics.

Sophie Aubin, T. H. (2006). Improving term extraction with terminological resources. In T. Salakoski, F. Ginter, S. Pyysalo, et T. Pahikkala (Eds.), *Advances in Natural Language Processing 5th International Conference on NLP, FinTAL 2006*, Turku, Finland, pp. 380–387. Springer.

Summary

This paper deals about semi-automatic induction of conceptual hierarchies from texts. Formal concept analysis seems to give good results in Cimiano et al. (2005) and more recently in Bedaoud et al. (2007). We show that on certain corpora (in particular from International Labour Office) results are less usable. Why ? We will test fca on different corpora and try to find why the results aren't as good as expected, focusing on a semi-automatic conceptualization task.

[4]Les extensions sont lemmatisées